While Loops

To this point, we have introduced "if" statements and "for" loops. Another important loop is a "while" loop. In a "while" loop, MATLAB keeps completing the MATLAB commands inside the loop until the governing logic statement becomes false.

Form

while <logic statement>
    MATLAB commands done if logic statement is true
end

Example

i=1; while i<5
    i=2*i;
end

Drop the semicolon that prevents output to the screen and watch what happens

After the completion of the above example, "i" would equal eight. Be careful when using while-loops because you can accidentally create an infinite loop, i.e. the end condition is never met.

Vectors

A vector is simply a matrix with a single row or a single column. Conveniently, MATLAB does have the ability to treat vectors differently than a regular matrix.

<table>
<thead>
<tr>
<th>MATLAB command</th>
<th>Result</th>
<th>The MATLAB default is to create a row vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;&gt; a = [5 6 -5]</td>
<td>(5 6 -5)</td>
<td></td>
</tr>
<tr>
<td>&gt;&gt; a(3) = 7</td>
<td>(5 6 7)</td>
<td></td>
</tr>
<tr>
<td>&gt;&gt; b(4) = -2</td>
<td>(0 0 0 -2)</td>
<td></td>
</tr>
</tbody>
</table>

Note: when using this vector notation, it does not matter whether or not the vector is a row or a column. MATLAB treats the two types of vectors the exact same.

Array Operators

MATLAB has the ability to perform operations between the components of matrices. For example, a user can multiply each component of a matrix A by the corresponding component of a matrix B to define a new matrix C. This is not the same as matrix multiplication. By placing a "," before an operator like *, ^ or /, MATLAB will perform the requested operation component by component.
Example:

```matlab
>> A = [1 2; 3 4];
>> B = [4 3; 2 1];
```

```matlab
>> C1 = A*B  % and the result
     C1 =
         8  5
        20 13
```

```matlab
>> C2 = A.*B  % Note: The matrices are not the same! Look at prob. #2.2 for an example use of an array operator
     C2 =
         4  6
        6  4
```

Script Files

So far, we have only introduced functions. Functions take a given input and return some output. For example, the congruent transformation function written for Laboratory #1 required two matrices as input and returned one matrix. The intermediate steps and variables required to create the output from the input were contained within the function and were not retrievable by the user. Namely, if a variable "temp" was created in a function, the user cannot recall "temp" at the MATLAB prompt. The variable "temp" exists solely inside the corresponding function. Therefore, multiple functions can use a variable named "temp" and there is not a problem.

Script files work differently than a function. Script files are just text files that contain a sequence of MATLAB commands. Specific inputs are not required nor are specific outputs returned. However, any variable used in the right-hand-side of an equation must be defined before that line of code is executed. Each MATLAB command is executed just as if the line was typed in at the MATLAB prompt. Thus, every variable created in a script file is retrievable at a MATLAB prompt. If a variable is created in the script file and a variable of the exact same name exists in the MATLAB memory buffer, the original variable will be erased.

<table>
<thead>
<tr>
<th>Form of a script</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>% any associated comments</td>
<td>% to call this script file, type the name of the script</td>
</tr>
<tr>
<td>%</td>
<td>% file without the .m extension at the MATLAB prompt</td>
</tr>
<tr>
<td>%</td>
<td>% A=[1 5; 3 8];</td>
</tr>
<tr>
<td>&lt;MATLAB commands&gt;</td>
<td>B=[4 -2; 2 1];</td>
</tr>
<tr>
<td></td>
<td>C=A*B;</td>
</tr>
</tbody>
</table>

To call the script named `<file name>.m`, type at the MATLAB prompt

```matlab
>> <file name>
```

To retrieve the matrices A, B, and C created in the script, type the matrix name at the MATLAB prompt. Script files can be useful in recreating a previous MATLAB session. By defining all your key variables in a script file, you can recreate these variables in a new session by simply running the script file at the beginning of the session.